

CLAIMS

We claim:

1. One or more computer-readable media having encoded thereon a
5 computer-readable data structure storing an intermediate representation of software, the
data structure comprising:

a plurality of instruction nodes representing a plurality of instructions of the
software;

- wherein the instruction nodes are operable to represent the instructions in a
10 machine-dependent manner and are further operable to represent the instructions in a
machine-independent manner.

2. The computer-readable media of claim 1 wherein instructions are
uniformly represented by a format for specifying at least the following for an
15 instruction:

an operator;

any number of or no destination operands associated with the operator via the
format; and

- any number of or no source operands associated with the operator via the
20 format.

3. The computer-readable media of claim 1 wherein instructions are
uniformly represented by a format for specifying an instruction node, zero or more
destination operand nodes, and zero or more source operand nodes.

25

4. The computer-readable media of claim 1 wherein the data structure
further comprises:

a plurality of operand nodes associated with the instruction nodes, wherein the
operand nodes represent a plurality of operands of the instructions of the software.

5. The computer-readable media of claim 4 wherein at least one data flow graph is threaded through the operand nodes.

5 6. The computer-readable media of claim 5 wherein the data flow graph comprises an SSA representation.

7. The computer-readable media of claim 4 wherein the operand nodes are further operable to be annotated to explicitly indicate at least one data flow graph for
10 the software.

8. The computer-readable media of claim 4 wherein the nodes of the data structure are further operable to store information explicitly indicating at least one data flow graph for the software without constructing a separate data structure therefor.
15

9. The computer-readable media of claim 4 wherein the nodes of the data structure are further operable to store information explicitly indicating at least one control flow graph for the software without constructing a separate data structure for the control flow graph.
20

10. The computer-readable media of claim 9 wherein the control graph for the software is indicated by associating at least one control flow operation to at least one target label via a control flow edge.

25 11. The computer-readable media of claim 9 wherein the control graph for the software is indicated by associating at least one exception causing instruction to at least one instance of exception handling code via a control flow edge.

12. The computer-readable media of claim 1 wherein the data structure further comprises:
a representation of non-instruction data of the software stored as an instruction.

5 13. The computer-readable media of claim 12 wherein the data structure further comprises:
a representation of instruction data of the software stored as a data instruction.

10 14. The computer-readable media of claim 1 wherein:
the data structure represents a lowered form of the software; and
at least one operand preserves type information specified in source code for the software.

15 15. The computer-readable media of claim 4 wherein:
at least one operand node is annotated with alias information.

16. One or more computer-readable media having encoded thereon a computer-readable data structure comprising:
an intermediate representation of software derived from source code; and
20 annotations for a plurality of analyses of the software, wherein a single format accommodates the annotations.

25 17. The computer-readable media of claim 16 wherein the intermediate representation of the software comprises a graph threaded through nodes of the representation.

18. The computer-readable media of claim 17 wherein the graph comprises a control flow graph representing control flow for the software.

19. The computer-readable media of claim 17 wherein the graph comprises a data flow graph representing data flow for the software.

20. One or more computer-readable media having encoded thereon a
5 computer-readable data structure storing an intermediate representation of software, the data structure comprising:

a plurality of instruction nodes representing a plurality of instructions of the software, wherein at least one of the instruction nodes represents a control flow instruction and at least one of the instruction nodes represents an opcode;

10 for at least one of the instruction nodes, one or more source operand nodes and one or more destination operand nodes, wherein at least one of the operand nodes represents a memory location and at least one of the operand nodes represents a label;

one or more links explicitly representing control flow for the software, wherein the control flow includes exception handling control flow;

15 one or more links explicitly representing data flow for the software;

information associated with at least one operand node indicating alias information for a variable associated with the operand node; and

at least one data instruction node of a same format as the instruction nodes but storing non-instruction data for the software.

20

21. The computer-readable media of claim 20 wherein the data structure is operable to represent the software in a machine-dependent and a machine-independent manner without changing format.

25

22. A system for software development, the system comprising:
means for representing software; and

means for analyzing the means for representing to analyze the software represented thereby;

wherein the means for representing is of a single format operable to represent the software in a machine-independent and a machine-dependent manner.

23. A method of processing a data structure storing an intermediate
5 representation of software for a compiler, wherein the intermediate representation is of a format, the method comprising:
conducting an analysis of the data structure; and
based on the analysis, taking an action without changing the format;
wherein the format is operable to represent the software in a machine-dependent
10 and a machine-independent manner.

24. The method of claim 23 wherein the action comprises:
annotating the intermediate representation with data.

- 15 25. The method of claim 23 wherein the annotating is done in situ.

26. The method of claim 23 wherein the action comprises:
generating information about the software.

- 20 27. The method of claim 26 further comprising:
producing code for the software by traversing the data structure and generating object code for instructions therein.

28. On one or more computer readable media, a software product generated
25 using the method of claim 26.

29. A method of processing a data structure encoded on one or more computer-readable media, wherein the data structure comprises a plurality of nodes, the method comprising:

starting at one of the nodes within the data structure; and

5 traversing to another of the nodes within the data structure;

wherein:

the data structure stores an intermediate representation of software;

the data structure comprises a plurality of nodes representing instructions of the software; and

10 at least one node representing an instruction is associated with one or more source operands and one or more destination operands.

30. The method of claim 29 wherein the data structure is operable to represent a machine-dependent and a machine-independent form of the software.

15

31. The method of claim 29 wherein the data structure is operable to represent operand types specified in source code in a low level representation of the software.

20 32. The method of claim 29 wherein the data structure is operable to explicitly represent control flow for the software.

33. The method of claim 32 wherein the control flow comprises exception handling, whereby the data structure is operable to explicitly represent control flow for exception handling of the software.

25

34. The method of claim 29 wherein the data structure is operable to explicitly represent data flow for the software.

35. The method of claim 29 wherein the data structure is operable to explicitly represent alias information for operands.

36. The method of claim 29 wherein the data structure comprises an association between a use of a variable and a definition of the variable.

37. The method of claim 29 further comprising:
based on analysis of the data structure, annotating the data structure with additional information.

38. The method of claim 37 wherein the annotating comprises associating an operand of one instruction with an operand of another instruction.

39. The method of claim 37 wherein the annotating comprises associating an instruction with another instruction.

40. The method of claim 29 further comprising:
based on analysis of the data structure, adding an instruction to the data structure.

41. The method of claim 29 further comprising:
based on analysis of the data structure, removing an instruction from the data structure.

42. The method of claim 29 further comprising:
based on analysis of the data structure, changing an instruction in the data structure.

43. On one or more computer-readable media, a software product generated using the method of claim 29.

5 44. A method of representing software, the method comprising:
representing each instruction and data element in a single format; and
representing each instruction as a data flow operation effected by execution of
the instruction.

10 45. The method of claim 44 wherein representing each instruction as a data
flow operation comprises explicitly representing side effects for the instruction.

46. A software development environment comprising:
one or more software development tools encoded on one or more computer-
readable media;
15 wherein the software development tools are operable to generate or analyze an
intermediate representation of software of a format operable to represent software in a
machine-independent and a machine-dependent manner.